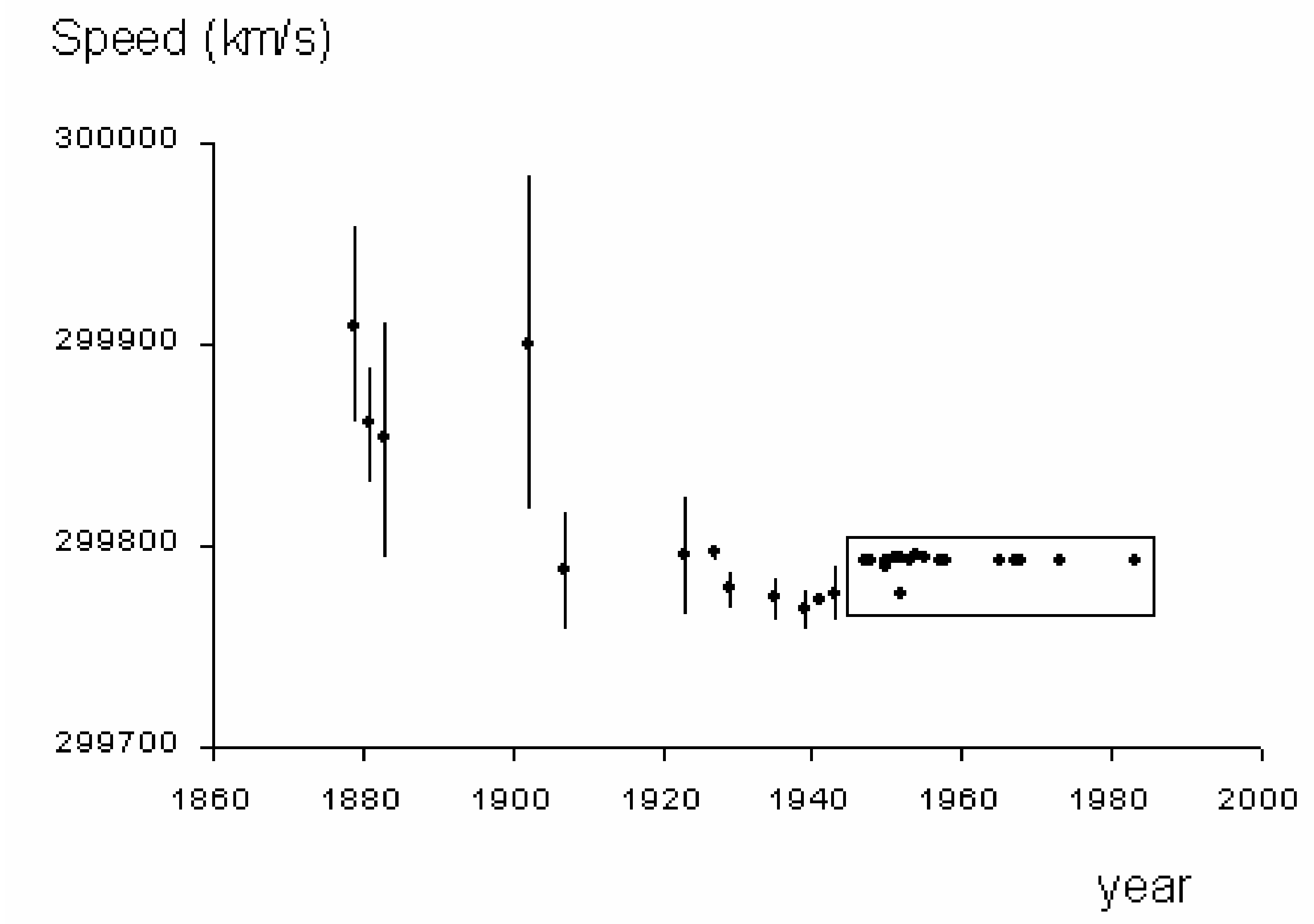
# Twelve Ways to Fool the Masses: Back to the Future

David H. Bailey
Lawrence Berkeley National Laboratory
http://crd.lbl.gov/~dhbailey

# Example from Physics: Measurements of Speed of Light



Why the discrepancy between pre-1945 and post-1945 values?
Probably due to biases and sloppy experimental methods.

## Example from the Social Sciences: The Blank Slate

The "Blank Slate" paradigm (1950-1990):

- The human mind at birth is a "blank slate."
- Heredity and biology play no significant role in human personality – all behavioral traits are socially constructed.

Current consensus, based on latest research:

- Humans at birth possess sophisticated facilities for language acquisition, pattern recognition and social life.
- Heredity and biology are the dominant factors of personality development.

How did these scientists get it so wrong?

- Sloppy experimental methodology and analysis.
- Pervasive biases and wishful thinking.

Ref: Steven Pinker, *The Blank Slate: The Modern Denial of Human Nature*

## History of Parallel Computing

- 1976-1986: Initial research studies and demos.
- 1986-1990: First large-scale systems deployed.
- 1990-1994: Successes over-hyped; faults ignored; shoddy measurement methods used; questionable performance claims made.
- 1994-1998: Numerous firms fail; agencies cut funds.
- 1998-2002: Reassessment.
- 2002-2006: Recovering?

## Parallel System Performance Practices, circa 1990

- Performance results on small-sized parallel systems were linearly scaled to full-sized systems.
  - Example:  8,192-CPU results were linearly scaled to 65,536-CPU results.
  - Rationale: “We can’t afford a full-sized system.”
  - Sometimes this was done without any clear disclosure in the paper or presentation.

## Parallel System Performance Practices, circa 1990

- Highly tuned programs were compared with untuned implementations on other systems.
  - In comparisons with vector systems, often little or no effort was made to tune the vector code.
  - This was the case even for comparisons with SIMD parallel systems – here the SIMD code can be directly converted to efficient vector code.

## Parallel System Performance Practices, circa 1990

- Inefficient algorithms were employed, requiring many more operations, in order to exhibit an artificially high Mflop/s rate.
  - Some scientists employed explicit PDE schemes for applications where implicit schemes were known to be much better.
  - One paper described doing a discrete Fourier transform by direct computation, rather than by using an FFT ($8n^2$ operations rather than $5n \log_2 n$).

## Parallel System Performance Practices, circa 1990

- Performance rates on 32-bit floating-point data on one system were compared with rates on 64-bit data on other systems.
  - Using 32-bit data instead of 64-bit data effectively doubles data bandwidth, thus yielding artificially high performance rates.
  - Some computations can be done safely with 32-bit floating-point arithmetic, but most cannot.
  - Even 64-bit floating-point arithmetic is not enough for some scientific applications – 128-bit is required.

## Parallel System Performance Practices, circa 1990

- In some cases, performance experiments reported in published results *were not actually performed*.
  - Abstract of published paper:

    "The current Connection Machine implementation runs at 300-800 Mflop/s on a full CM-2, or at the speed of a single processor of a Cray-2 on 1/4 of a CM-2."
  - Buried in text:

    "This computation requires 568 iterations (taking 272 seconds) on a 16K Connection Machine."

    i.e., the computation was not run on a full 64K CM-2.

    "In contrast, a Convex C210 requires 909 seconds to compute this example.  Experience indicates that for a wide range of problems, a C210 is about 1/4 the speed of a single processor Cray-2, …"

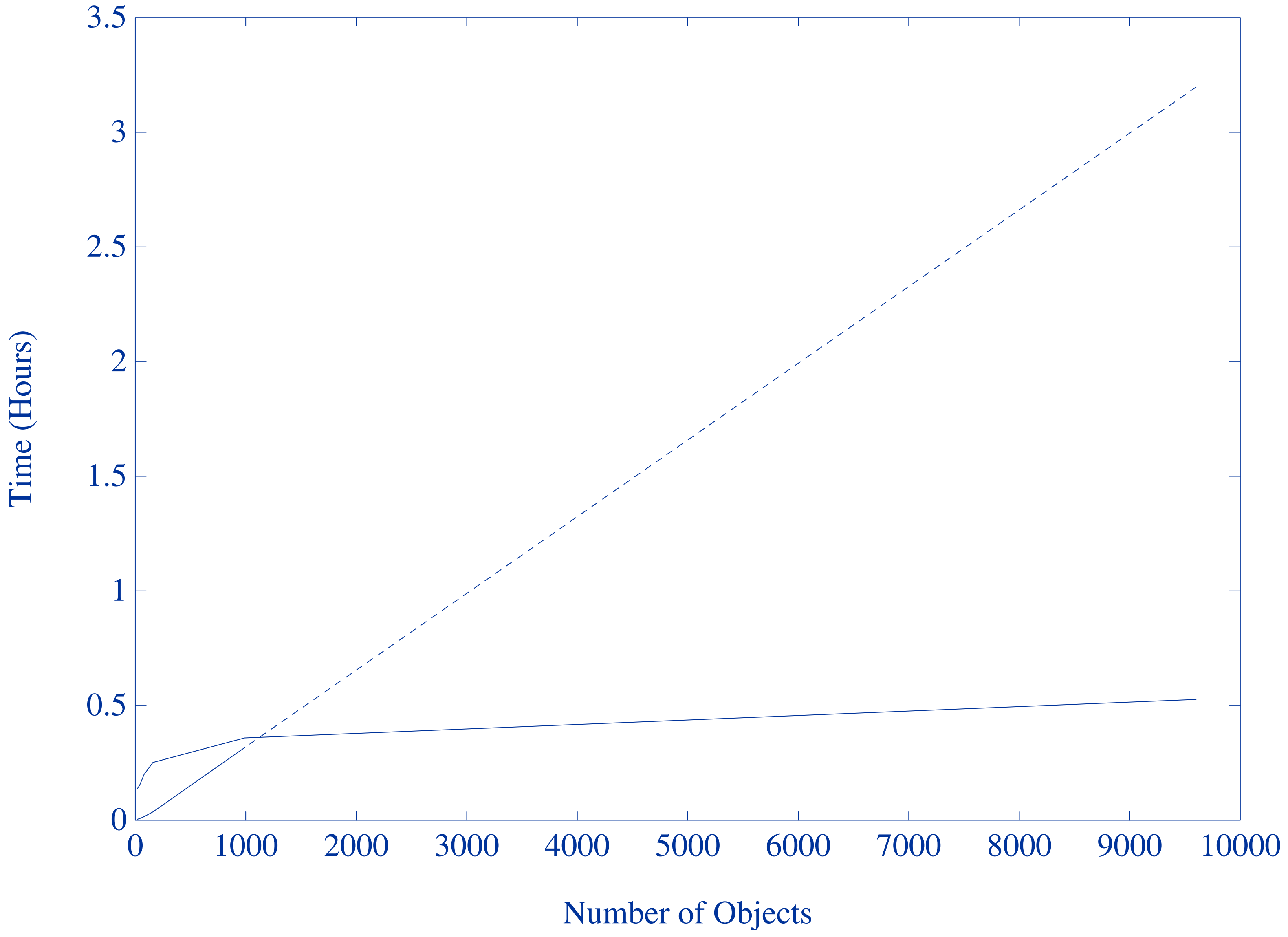
    i.e., the computation was not run on a Cray-2 at all.

## Parallel System Performance Practices, circa 1990

- Scientists were just as guilty as commercial vendors of questionable performance reporting.
  - The examples in my files were written by professional scientists and published in peer-reviewed journals and conference proceedings.
  - One example is from an award-winning paper.
- Scientists in some cases accepted free computer time or research funds from vendors, but did not disclose this fact in their papers.

Scientists should be held to a higher standard than vendor marketing personnel.

# Performance Plot A



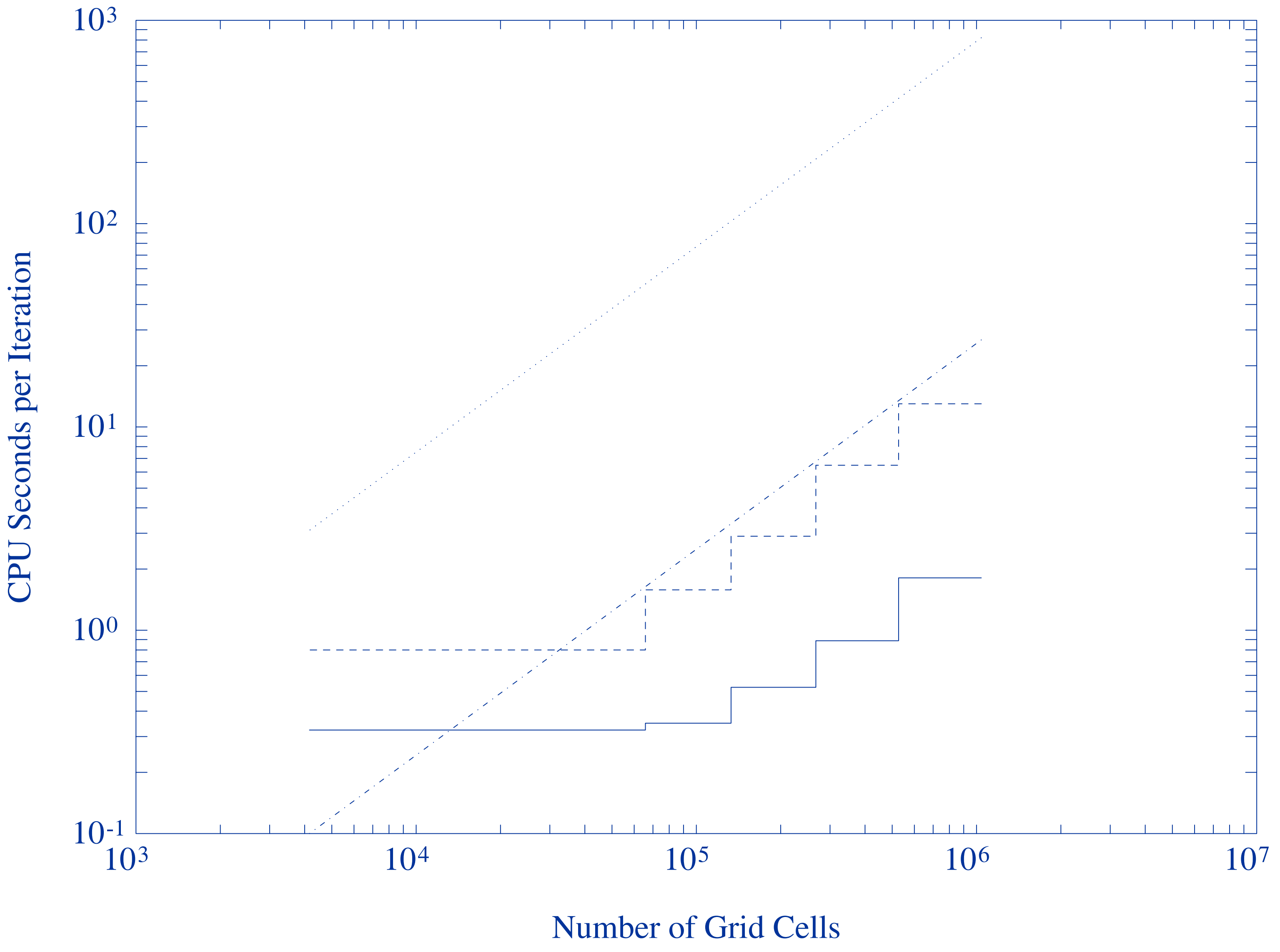
## Data for Plot A

| Total Objects | Parallel system Run time | Vector system Run time |
|---|---|---|
| 20 | 8:18 | 0:16 |
| 40 | 9:11 | 0:26 |
| 80 | 11:59 | 0:57 |
| 160 | 15:07 | 2:11 |
| 990 | 21:32 | 19:00 |
| 9600 | 31:36 | 3:11:50* |

**Notes:**

- In last entry, the 3:11:50 figure is an estimate.
- The vector system code is “not optimized.”
- The vector system performance is better except for the last (estimated) entry.

# Performance Plot B



# Facts for Plot B

- 32-bit performance rates on a parallel system are compared with 64-bit performance on a vector system.
- Parallel system results are linearly extrapolated to a full-sized system from a small system (only 1/8 size).
- The vector version of code is “unvectorized.”
- The vector system “curves” are straight lines – i.e., they are linear extrapolations from a single data point.

**Summary:**

It appears that of all points on four curves in this plot, at most four points represent real timings.

## Twelve Ways to Fool the Masses

1. Quote only 32-bit performance results, not 64-bit results.
2. Present performance figures for an inner kernel, and then represent these figures as the performance of the entire application.
3. Quietly employ assembly code and other low-level language constructs.
4. Scale up the problem size with the number of processors, but omit any mention of this fact.
5. Quote performance results projected to a full system.
6. Compare your results against scalar, unoptimized code on conventional systems.

## Twelve Ways to Fool the Masses

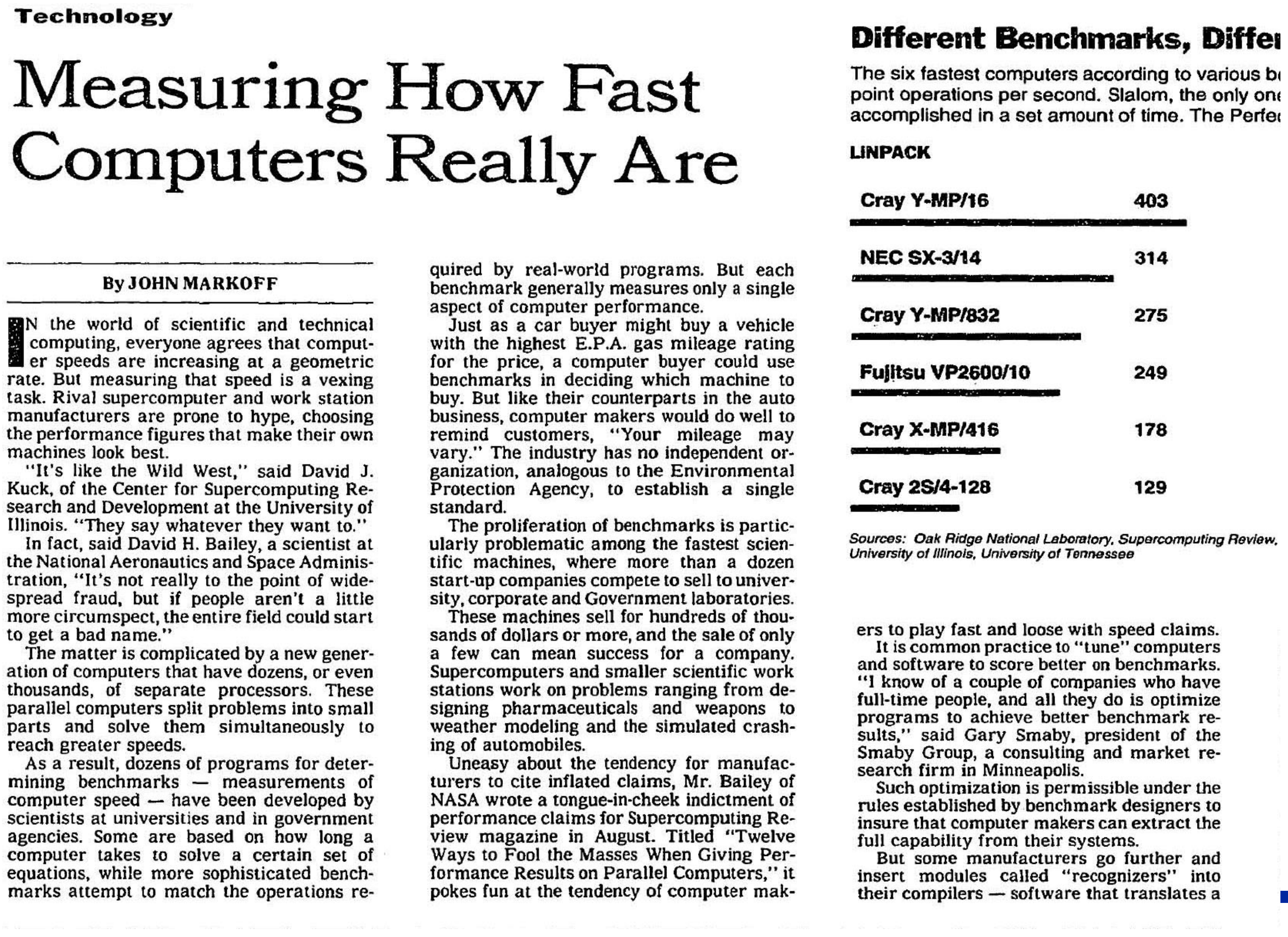
7. When direct run time comparisons are required, compare with an old code on an obsolete system.
8. If Mflop/s rates must be quoted, base the operation count on the parallel implementation, not on the best sequential implementation.
9. Quote performance in terms of processor utilization, parallel speedups or Mflop/s per dollar.
10. Mutilate the algorithm used in the parallel implementation to match the architecture.
11. Measure parallel run times on a dedicated system, but measure conventional run times in a busy environment.
12. If all else fails, show pretty pictures and animated videos, and don't talk about performance.

## Twelve Ways: Basic Principles

- Use well-understood, community-defined metrics.
- Cite performance rates based on efficient algorithms, not on schemes that exhibit artificially high Mflop/s rates.
- Use comparable levels of tuning.
- Provide full details of experimental environment, so that performance results can be reproduced by others.
- Disclose any details that might affect an objective interpretation of the results.
- Honesty and reproducibility should characterize all work.

**Danger:** We can fool ourselves, as well as others.

# New York Times, 22 Sept 1991



Technology

## Measuring How Fast Computers Really Are

By JOHN MARKOFF

IN the world of scientific and technical computing, everyone agrees that computer speeds are increasing at a geometric rate. But measuring that speed is a vexing task. Rival supercomputer and work station manufacturers are prone to hype, choosing the performance figures that make their own machines look best.

"It's like the Wild West," said David J. Kuck, of the Center for Supercomputing Research and Development at the University of Illinois. "They say whatever they want to."

In fact, said David H. Bailey, a scientist at the National Aeronautics and Space Administration, "It's not really to the point of widespread fraud, but if people aren't a little more circumspect, the entire field could start to get a bad name."

The matter is complicated by a new generation of computers that have dozens, or even thousands, of separate processors. These parallel computers split problems into small parts and solve them simultaneously to reach greater speeds.

As a result, dozens of programs for determining benchmarks — measurements of computer speed — have been developed by scientists at universities and in government agencies. Some are based on how long a computer takes to solve a certain set of equations, while more sophisticated benchmarks attempt to match the operations required by real-world programs. But each benchmark generally measures only a single aspect of computer performance.

Just as a car buyer might buy a vehicle with the highest E.P.A. gas mileage rating for the price, a computer buyer could use benchmarks in deciding which machine to buy. But like their counterparts in the auto business, computer makers would do well to remind customers, "Your mileage may vary." The industry has no independent organization, analogous to the Environmental Protection Agency, to establish a single standard.

The proliferation of benchmarks is particularly problematic among the fastest scientific machines, where more than a dozen start-up companies compete to sell to university, corporate and Government laboratories.

These machines sell for hundreds of thousands of dollars or more, and the sale of only a few can mean success for a company. Supercomputers and smaller scientific work stations work on problems ranging from designing pharmaceuticals and weapons to weather modeling and the simulated crashing of automobiles.

Uneasy about the tendency for manufacturers to cite inflated claims, Mr. Bailey of NASA wrote a tongue-in-cheek indictment of performance claims for Supercomputing Review magazine in August. Titled "Twelve Ways to Fool the Masses When Giving Performance Results on Parallel Computers," it pokes fun at the tendency of computer makers to play fast and loose with speed claims.

It is common practice to "tune" computers and software to score better on benchmarks. "I know of a couple of companies who have full-time people, and all they do is optimize programs to achieve better benchmark results," said Gary Smaby, president of the Smaby Group, a consulting and market research firm in Minneapolis.

Such optimization is permissible under the rules established by benchmark designers to insure that computer makers can extract the full capability from their systems.

But some manufacturers go further and insert modules called "recognizers" into their compilers — software that translates a

**Different Benchmarks, Diffe**

The six fastest computers according to various b
point operations per second. Slalom, the only one
accomplished in a set amount of time. The Perfe

LINPACK

| | |
|---|---|
| Cray Y-MP/16 | 403 |
| NEC SX-3/14 | 314 |
| Cray Y-MP/832 | 275 |
| Fujitsu VP2600/10 | 249 |
| Cray X-MP/416 | 178 |
| Cray 2S/4-128 | 129 |

*Sources: Oak Ridge National Laboratory, Supercomputing Review, University of Illinois, University of Tennessee*

## Excerpts from NYT Article

“Rival supercomputer and work station manufacturers are prone to hype, choosing the performance figures that make their own systems look better.”

“It’s not really to the point of widespread fraud, but if people aren’t somewhat more circumspect, it could give the field a bad name.”

## Fast Forward to 2004: Five New Ways to Fool the Masses

- Dozens of runs are made, but only the best performance figure is cited in the paper.
- Runs are made on part of an otherwise idle system, placing an unrealistically light load on the network.
- Performance rates are cited for a run with only one CPU active per node.
- Special operating system or compiler settings are used that are not appropriate for real-world usage.
- “Scalability” is defined as a successful execution on a large number of CPUs, regardless of performance.

**And lots of:** “Show pretty pictures and animated videos, and don't talk about performance.”

# Grid Computing Projects

BERKELEY LAB

NEESgrid

iVDgL

European GRID Forum

BIRN

NPACI GRID

DOE SciEnce Grid

www.doesciencegrid.org

Data GRID

e-Science dti

TERAGRID

EUROGRID

HIGH-PERFORMANCE COMPUTATION
BROADBAND NETWORK CONNECTIVITY
PARTNERSHIP
PEOPLE AND TRAINING
LARGE DATABASES DIGITAL LIBRARIES
INSTRUMENTATION

NSF Cyberinfrastructure

NVO

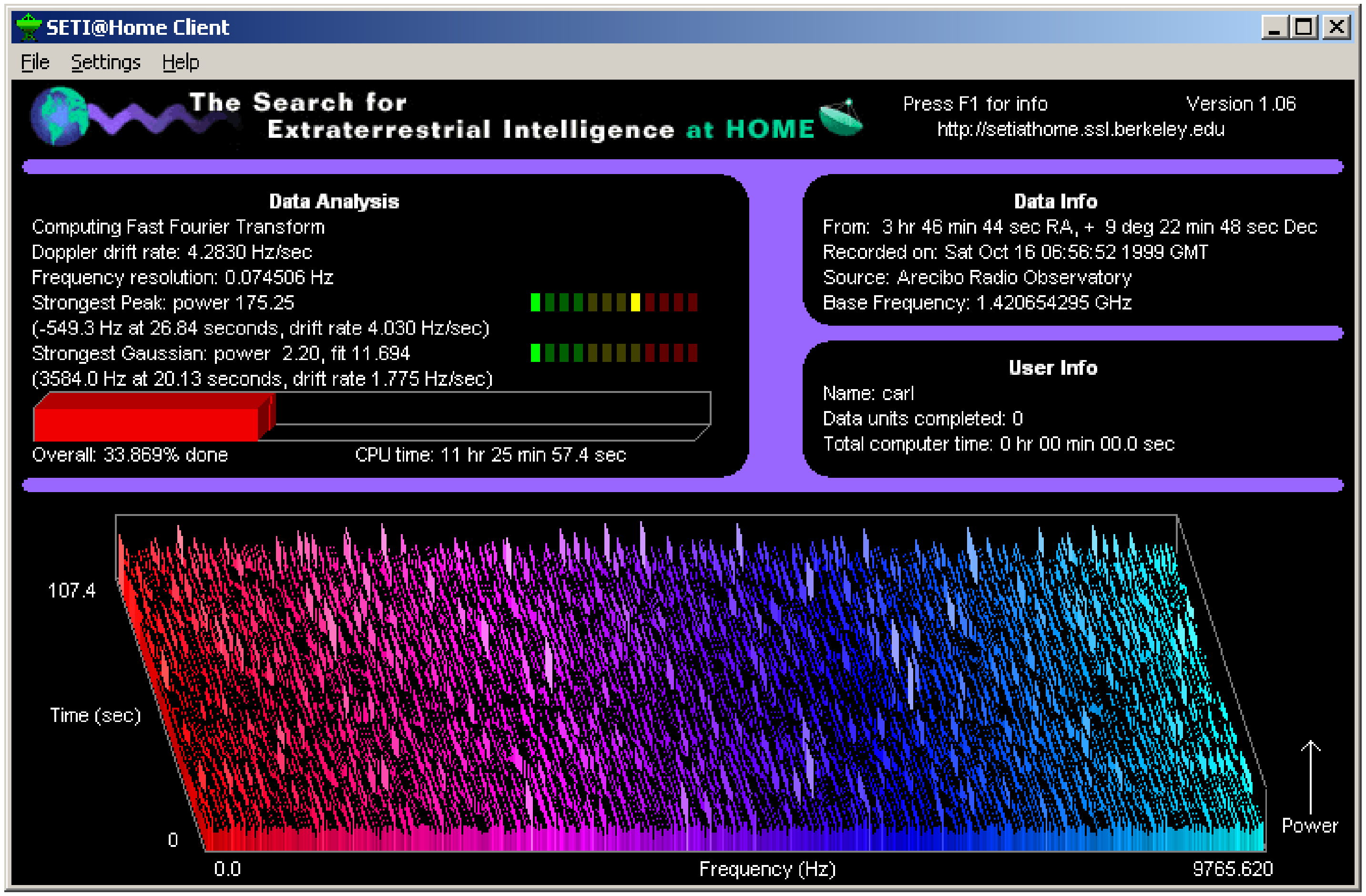
NATIONAL VIRTUAL OBSERVATORY

GEON

GLOBAL GF

PRAGMA

# SETI@Home



**Data Analysis**
Computing Fast Fourier Transform
Doppler drift rate: 4.2830 Hz/sec
Frequency resolution: 0.074506 Hz
Strongest Peak: power 175.25
(-549.3 Hz at 26.84 seconds, drift rate 4.030 Hz/sec)
Strongest Gaussian: power 2.20, fit 11.694
(3584.0 Hz at 20.13 seconds, drift rate 1.775 Hz/sec)
Overall: 33.869% done
CPU time: 11 hr 25 min 57.4 sec

**Data Info**
From: 3 hr 46 min 44 sec RA, + 9 deg 22 min 48 sec Dec
Recorded on: Sat Oct 16 06:56:52 1999 GMT
Source: Arecibo Radio Observatory
Base Frequency: 1.420654295 GHz

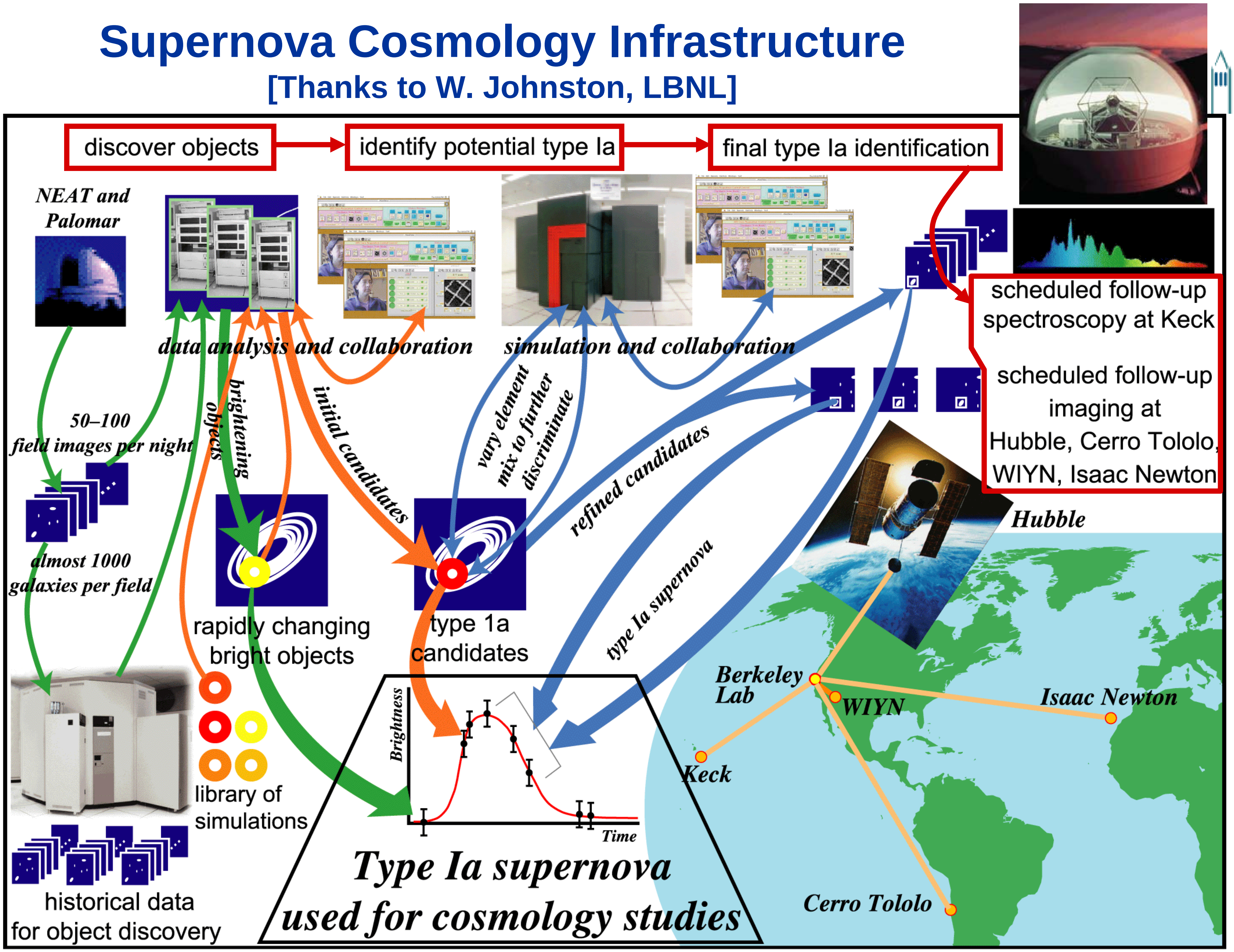
**User Info**
Name: carl
Data units completed: 0
Total computer time: 0 hr 00 min 00.0 sec

Seti@home sustains 35 Tflop/s on 2M+ systems

1.7 x $10^{21}$ flops over 3 years

# Supernova Cosmology Infrastructure

## [Thanks to W. Johnston, LBNL]



discover objects
identify potential type Ia
final type Ia identification
NEAT and Palomar
data analysis and collaboration
simulation and collaboration
50–100 field images per night
almost 1000 galaxies per field
brightening objects
initial candidates
vary element mix to further discriminate
refined candidates
type Ia supernova
scheduled follow-up spectroscopy at Keck
scheduled follow-up imaging at Hubble, Cerro Tololo, WIYN, Isaac Newton
Hubble
rapidly changing bright objects
type 1a candidates
library of simulations
historical data for object discovery
Brightness
Time
Type Ia supernova used for cosmology studies
Berkeley Lab
WIYN
Isaac Newton
Keck
Cerro Tololo

Supernova Cosmology Project, Perlmutter, et al. (http://www.supernova.lbl.gov)

## Potential for Overselling the Grid

- “All supercomputer computations will soon be done on grids.”
- “With the grid, every scientist will have access to all scientific data.”
- “A computational grid has greater capacity than its constituent systems.”
- “Corporate data processing will soon be handled by SETI-at-home-style computing utilities.”
- Etc.

## What the Grid Does Well

- Providing national or international access to important scientific datasets.
- Providing a uniform scheme for remote system access and user authentication.
- Providing a high-performance parallel platform for certain very loosely coupled computations.
- Providing a high-capability platform for large computations that can run on a single remote system, chosen at run time.
- Enabling new types of multi-disciplinary, multi-system, multi-dataset research.

## What the Grid Doesn’t Do So Well

- Scientific computations that require heavy interprocessor communication.
    - Probably the majority of high-end scientific computations are of this nature.
    - This doesn’t rule out such applications running remotely on a single system connected to the grid.
- Many classified or proprietary computations.
    - Current grid security and privacy are not convincing for many of these users
    - This doesn’t rule out “internal grids” -- some have been quite successful.

## Combating Performance Abuse: The Role of Intelligent Benchmarks

- Well-designed, rigorous, scalable performance benchmark tests.
  - Must be produced by a community-based effort.
  - Must be a based on codes that have credibility as a “useful” scientific or commercial application.
  - Must be easily implemented without lengthy, highly expert effort.
  - Must be appropriate for moderate-sized systems as well as very large systems.
  - Must include a clear path to increase problem sizes for future use.

## The Role of Intelligent Benchmarks

- Well-thought-out “ground rules.”
  - How much tuning of the benchmark is permitted?
  - How is the extent of tuning measured?
  - How will disputes be settled?

If ground rules can be abused, they will be abused.

## The Role of Intelligent Benchmarks

- A rational scheme for calculating performance rates.
  - How is run time measured?
  - Is required initialization included in the run time?
  - How will operation counts or the amount of work performed be reckoned?
- A well-defined test to validate the correctness of the results.
  - It is best if the benchmark includes its own scalable validity test.
  - At the least, spot checks of results are needed.

## The Role of Intelligent Benchmarks

- A well-supported repository of results.
  - Kept up to date – a long-term commitment.
  - Includes all environmental and system information.
  - New results periodically solicited.
  - A searchable database is preferred.

## Twelve Ways: Back to the Future

- Use well-understood, community-defined metrics.
- Cite performance rates based on efficient algorithms, not on schemes that exhibit artificially high Mflop/s rates.
- Use comparable levels of tuning.
- Provide full details of experimental environment, so that performance results can be reproduced by others.
- Disclose any details that might affect an objective interpretation of the results.
- Honesty and reproducibility should characterize all work.

**Danger:** We can fool ourselves, as well as others.